UTILITY APPLICATION

OF

CHI-PEI M. HSING AND ALAN T. YAUNG

FOR

UNITED STATES LETTERS PATENT

ON

SECURE ACCESS TO A UNIFIED LOGON-ENABLED DATABASE SERVER

Docket No.: ST9-99-167 (IBMST 43948)

Drawings: 4

Attorneys
PRETTY, SCHROEDER & POPLAWSKI
444 South Flower Street, 19th Floor
Los Angeles, California 90071
Ofc: 213/622-7700
Fax: 213/489-4210

CERTIFICATE OF MAILING BY "EXPRESS MAIL"

"EXPRESS MAIL" MAILING LABEL NUMBER EL3449967764US

DATE OF DEPOSIT February 24, 2000

I HEREBY CERTIFY THAT THIS PAPER OR FEE IS BEING DEPOSITED WITH THE UNITED STATES POSTAL SERVICE "EXPRESS MAIL POST OFFICE TO ADDRESSEE" SERVICE UNDER 37 CFR-10 ON THE DATE INDICATED ABOVE AND IS ADDRESSED TO BOX PATENT APPLICATION, THE ASSISTANT COMMISSIONER FOR PATENTS, WASHINGTON, D. C., 20231.

Sherlin Yaghoubzadeh

(TYPED OR PRINTED NAME OR PERSON MAILING PAPER OR FEE)

(SIGNATURE OF PERSON MAILING PAPER OR FEE)

SECURE ACCESS TO A UNIFIED LOGON-ENABLED DATABASE SERVER

BACKGROUND OF THE INVENTION

1. Field of the Invention.

This invention relates in general to computer implemented database management systems, and more particularly, to a technique for secure access to a unified logon-enabled database server.

2. Description of Related Art.

A data store is a term used to refer to a generic data storage facility, such as a relational database, flat-file, hierarchical database, etc. For nearly half a century computers have been used by businesses to manage information such as numbers and text, mainly in the form of coded data. However, business data represents only a small part of the world's information. As storage, communication and information processing technologies advance, and as their costs come down, it becomes more feasible to digitize other various types of data, store large volumes of it, and be able to distribute it on demand to users at their place of business or home.

New digitization technologies have emerged in the last decade to digitize images, audio, and video, giving birth to a new type of digital multimedia information. These multimedia objects are quite different from the business data that computers managed in the past, and often require more advanced information management system infrastructures with new capabilities. Such systems are often called "digital libraries."

Databases are computerized information storage and retrieval systems. For example, a Relational Database Management System (RDBMS) is a database management system (DBMS) which uses relational techniques for storing and retrieving data. Relational databases are organized into physical tables which consist of rows and columns of data. The rows are formally called tuples. A database will typically have many physical tables and each physical table will typically have multiple tuples and multiple columns. The physical tables are typically stored on

random access storage devices (RASED) such as magnetic or optical disk drives for semi-permanent storage. Additionally, logical tables or "views" can be generated based on the physical tables and provide a particular way of looking at the database. A view arranges rows in some order, without affecting the physical organization of the database.

In some systems, a client computer is networked to a server computer, for example via a computer network. The client computer may use a user management function provided by its operating system (e.g., Windows NT®) to control its logon process. For example, the Windows NT® operating system requires a user to enter a user name and a password to use the client computer. In a unified logon environment, each client computer connecting to a database server computer needs to have a corresponding user identifier (ID) and password created on the server computer.

This unified logon requirement creates an administrator's nightmare because the administrator needs to maintain all of the client user name/password and server user ID/password combinations for each user and for each client and server computer. In some systems, there could be more than a thousand client computers connecting to the same server computer, requiring the maintenance of thousands of user name/password or user ID/password combinations. Maintaining user names, user identifiers and passwords may be even more complicated in a federated system in which many client computers are connected to many server computers that are connected together. Furthermore, to maintain a high level of security, many companies are now resorting to changing passwords every three to six months, which adds to the administrative burden, as well as being a burden on users.

In particular, a system administrator may manually change the passwords at the server computer, and then the system administrator notifies the users of their new passwords. There is a security risk in that the new password from the administrator to users could be intercepted by hackers. The same situation applies to users, when they manually change their passwords themselves and then inform their administrator of the new passwords.

Thus, there is a need in the art for an improved technique for secure access to a unified logon-enabled database server.

SUMMARY OF THE INVENTION

To overcome the limitations in the prior art described above, and to overcome other limitations that will become apparent upon reading and understanding the present specification, the present invention discloses a method, apparatus, and article of manufacture for a computer-implemented technique for secure access to a unified logon-enabled database server.

In accordance with the present invention, security is provided for a computer connected to a data store. Initially, an authentication key, a user name, and a computer identifier are received. The authentication key is parsed to obtain a parsed user name and computer identifier. The parsed user name and computer identifier are validated using the received user name and computer identifier.

Additionally, in accordance with yet another embodiment, if the received user name and computer identifier are validated, the authentication key is parsed to obtain a server user identifier and a server password. Then, the parsed server user identifier and server password are used to connect to a database server computer.

BRIEF DESCRIPTION OF THE DRAWINGS

Referring now to the drawings in which like reference numbers represent corresponding parts throughout:

FIG. 1 is a schematic diagram illustrating a hardware environment of an embodiment of the present invention, and more particularly, illustrates a typical distributed computer system;

FIG. 2 illustrates a table that summarizes the structure for an authentication key before encryption;

FIG. 3 is a flow diagram illustrating the steps performed by the Secure Access System to generate and forward an authentication key; and

FIG. 4 is a flow diagram illustrating the process of using an authentication key.

DETAILED DESCRIPTION

In the following description of an embodiment of the invention, reference is made to the accompanying drawings which form a part hereof, and which is shown by way of illustration a

specific embodiment in which the invention may be practiced. It is to be understood that other embodiments may be utilized as structural changes may be made without departing from the scope of the present invention.

Hardware Environment

FIG. 1 schematically illustrates the hardware environment of an embodiment of the present invention, and more particularly, illustrates a typical distributed computer system using the network 100 to connect client computers 102 executing client applications to a server computer 104 executing software and other computer programs, and to connect the server system 104 to data sources 106. In one embodiment, the client computers 102 are Windows NT® workstations, and the server computer 104 is a Universal Database (UDB) server computer. It is to be understood that the Windows NT® workstations and Universal Database Server are being used for illustration only, and the invention may be practiced with other databases or computers.

A Secure Access System 110 may reside on another computer 108 that is connected to the network 100. Although the Secure Access System 110 is shown at a computer 108 separate from the client computers 102 and the server computer 104, it can be envisioned that the Secure Access System 110 may reside on a client computer 102, the server computer 104, or some combination of computers.

A typical combination of resources may include client computers 102 that are personal computers or workstations, and a server computer 104 that is a personal computer, workstation, minicomputer, or mainframe. These systems are coupled to one another by various networks, including LANs, WANs, and the Internet. The data sources 106 may be geographically distributed.

A client computer 102 typically executes a client application and is coupled to a server computer 104 executing server software. The client application program is typically a software program which can include, *inter alia*, multi-media based applications, e-mail applications, e-business applications, or workflow applications. The server software is typically a program such as DB2 Universal Database (UDB)® from International Business Machines, Corporation. The

server computer 104 also uses a data source interface and, possibly, other computer programs, for connecting to the data sources 106. The client computer 102 is bi-directionally coupled with the server computer 104 over a line or via a wireless system. In turn, the server computer 104 is bi-directionally coupled with data sources 106. The computer 110 is bidirectionally coupled with the client computers 102 and the server computers 104. In one embodiment, the Secure Access System 110 intercepts data from the client computer to the server computer and performs security processing, as will be discussed below.

The computer programs executing at each of the computers, including the Secure Access System 110, are comprised of instructions which, when read and executed by the computers, cause the computers to perform the steps necessary to implement and/or use the present invention. Generally, computer programs are tangibly embodied in and/or readable from a device, carrier, or media, such as memory, data storage devices, and/or data communications devices Under control of an operating system, the computer programs may be loaded from the memory, data storage devices, and/or data communications devices into the memory of each computer for use during actual operations.

Thus, the present invention may be implemented as a method, apparatus, system, or article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. The term "article of manufacture" (or alternatively, "computer program product") as used herein is intended to encompass a computer program accessible from any computer-readable device, carrier, or media, including the internet. Of course, those skilled in the art will recognize many modifications may be made to this configuration without departing from the scope of the present invention.

Those skilled in the art will recognize that the environment illustrated in FIG. 1 is not intended to limit the present invention. Indeed, those skilled in the art will recognize that other alternative hardware environments may be used without departing from the scope of the present invention.

SECURE ACCESS SYSTEM

Due to the rapid growth in the use of client/server systems, security issues for these systems have become increasingly important. The Secure Access System 110 will address two

problems associated with a unified logon-enabled database server computer, namely, password maintenance and password security.

The Secure Access System 110 reduces the number of server user ID/password combinations needed by a unified logon-enabled database server computer in a secure environment. With the Secure Access System 110, the system administrator only needs to create a few server user IDs on a server computer and assign access privileges to these user IDs individually. These server user IDs will be used by a number of users in an operating system with the unified logon capability. For example, a server user ID may be assigned to five client workstation users, allowing each of them to connect to a server computer with the same user ID.

The Secure Access System 110 consists of four elements and processes: (1) the structure of an authentication key, (2) the procedure to generate an authentication key, (3) the delivery of the authentication key, and (4) the procedure to parse an authentication key. Based on the structure defined in the part (1), a generator program of the Secure Access System 110 processes part (2) and part (3). A parser program of the Secure Access System 110 handles part (4).

Structure of An Authentication Key

An authentication key is comprised of four pieces of information:

- server (e.g., UDB) user ID,
- server (e.g., UDB) password,
- client (e.g., Windows NT®) user name, and
- client (e.g., Windows NT® computer) IP address.

The server user ID and password are used to log onto the server computer. The client user name is the user name at the client workstation. The client IP address refers to an identifier of a computer (i.e., a computer identifier) in a TCP/IP network. When a user initially is assigned to a client workstation, the user provides the system administrator with a client user name and an IP address of the client workstation. A server user ID and a server password are assigned to the user by a system administrator. Additionally, the user provides an E-mail address to the system administrator.

In terms of addressing the concern of password security, even though a group of client users use the same server user ID/password to connect to a server computer, the authentication keys they use are unique. In fact, even two users from the same client workstation will have different keys. Moreover, a stolen authentication key is useless, since it can only be used from a specific client workstation with a specific client user name (which is obtained when a user logs onto a client workstation with a client user name and password).

As stated above, an authentication key is constructed from four pieces of information: a server user ID, a server password, a client user name, and an IP address of a client workstation. The structure of an authentication key is described below:

1. All the characters in the authentication key are readable/printable.

2. IP address is represented in dotted decimal notation, which has 4 strings separated by '.' (e.g., 128.10.2.30). Each string will be extracted and padded with leading zero's to make it a 3-digit string. For example, 128.10.2.30 is converted to 128.010.002.030.

3. An authentication key has nine fields:
   a. The first field stores the fourth string of the IP address, and its length is 3-byte.
   b. The second field holds the server user ID, and its length is variable.
   c. The third field has 1-byte length for a separator ','.
   d. The fourth field stores the third string of the IP address, and its length is 3-byte.
   e. The fifth field stores the server password, and its length is variable.
   f. The sixth field has 1-byte length for a separator ','.
   g. The seventh field contains the second string of the IP address, and its length is 3-byte.
   h. The eighth field holds the client user name, and its length is variable.
   i. The ninth field contains the first string of the IP address, and its length is 3-byte.

FIG. 2 illustrates a table 200 that summarizes the structure for an authentication key before encryption (i.e., in an unencrypted format). In particular, the table 200 illustrates the field and length for each of the nine fields of an authentication key.

For example, if the server user ID is ADMINUSER, the server password is HIGHPRIV, client user name is Michael, and the client IP address is 9.112.19.75, the authentication key in an unencrypted format is as follows:

075ADMINUSER,019HIGHPRIV,112Michael009

Generating an Authentication Key

This authentication key is generated by the generator program of the Secure Access System 110 as an encrypted, printable string, which is further described in U.S. Patent Application Serial No. 09/397,439, filed on September 17, 1999, entitled A TECHNIQUE OF PASSWORD ENCRYPTION AND DECRYPTION FOR USER AUTHENTICATION IN A FEDERATED CONTENT MANAGEMENT SYSTEM, by Michael C. Hsing, et al., which is incorporated by reference herein. Based on the encryption technique of U.S. Patent Application Serial No. 09/397,439, every eight characters will be encrypted into a 12-byte string. The remaining characters with a length less than eight will also be encrypted into a 12-byte string.

Continuing with the example, the authentication key (38 bytes in length) for Michael before encryption is arranged into rows of eight characters (for viewing convenience):

075ADMIN
USER, 019
HIGHPRIV
,112Mich
ael1009

This is a combination of the IP address (9.112.19.75) for a client workstation, a server user ID, a server password, and a client user name. The encrypted string of this authentication key (60 bytes in length) is shown below as rows of twelve characters (for viewing convenience):

8,S"e]=-rDDF
###n"Y66`tP@
uqVd#%Xad@D@
_v =""8\`fr@
Hk |v`*"(`DBR

Delivery of the Authentication Key

The Secure Access System 110 maintains a notification list of users, where each entry in the list includes a client user name, a client workstation name, and an E-mail address. Once the authentication keys are generated and encrypted, the Secure Access System 110 forwards the authentication keys to the E-mail addresses identified in the notification list.

The system administrator may maintain a notification list as follows:

Client User Name	Client Workstation Name	E-mail address
Michael	chsing	mhsing@us.ibm.com
. . .		

Then, the encrypted authentication key below is sent to the E-mail address: mhsing@us.ibm.com.

8, S"e]=-rDDF###n"Y66 ` tP@uqVd#%Xad@D@_v =""8\`fr@Hk |v`*"(`DBR

Parsing an Authentication Key

When a user logs onto a client workstation, the client workstation stores the user name. Then, the client workstation connects to the server computer. In particular, when connecting to a server computer from a client workstation, the client workstation transmits the authentication key to the server computer. Additionally, when the authentication key is transmitted, the client workstation also transmits the *actual IP address and client user name*.

The parser program of the Secure Access System 110 intercepts the authentication key after it has been sent by the client workstation and before it is received by the server computer. The parser program of the Secure Access system 110 will process the authentication key and decrypt the authentication key into an IP address, a client user name, a server user ID, and a server password. If the decrypted IP address and client user name match the actual IP address and client user name, the parser program of the Secure Access System 110 uses the decrypted server user ID and server password to connect to the server computer. Otherwise, the Secure Access System 110 returns an error. Moreover, if an authentication key cannot be found, the default client workstation logon user ID and password will be used.

This technique addresses the concerns of password maintenance. When a password expires on a server computer, the system administrator will simply use the Secure Access System 110 to send out new authentication keys to each user. The administrative burden is definitely relieved.

Unified logon is an important requirement for today's enterprise computing. However, it imposes additional administrative burdens on a system administrator and introduces potential security problems. The Secure Access System 110 provides a secure approach to accessing a unified logon-enabled database server computer. The Secure Access System 110 reduces the administrative burden of maintaining passwords and enhances the security of passwords. The Secure Access System 110 provides significant benefits to client/server systems, in which a server computer is used for storing administrative information for accessing, for example, a federated data store that is linked to various back-end server computers, such as IBM's Digital Library, Visuallnfo 400, ImagePlus 390, Domino.Doc, etc.

One unique aspect of the invention is the concept of an authentication key. Without a correct authentication key, a user ID/password is totally useless. This provides additional security for the database server computer.

Scenarios for Using Authentication Key

The system administrator may maintain a notification list like this:

Client User Name	Client Workstation Name	E-Mail address
Michael	chsing	mhsing@us.ibm.com
Alan	ayaung	ayaung@us.ibm.com
...	...	...

The administrator creates a user: ADMINUSER with password:HIGHPRIV on the server computer and grants it the proper access privileges. If the administrator wants to let all the users (e.g., Michael, Alan, etc.) in the list access the server computer as ADMINUSER/HIGHPRIV, the system administrator will run a generator program to generate and send an unique authentication key to each user on the list.

The authentication key for Michael before encryption is as follows:

075ADMINUSER, 019HIGHPRIV, 112Michael009

The encrypted string of this authentication key, which can be sent to the user through E-mail, is as follows:

8,S"e]=-rDDF###n"Y66`tP@uqVd#%Xad@D@_v =""8\`fr@Hk |v`*"(`DBR

By the same token, the authentication key for user Alan (IP address 9.112.21.225) before encryption is as follows:

225ADMINUSER,021HIGHPRIV,112Alan009

After encryption, the authentication key is as follows:

+5y-hTJATB@`0 |6s-LV6P`D`uqVd#%Xad@D@;i(IO9&QP@Bd" m)&r%(`FTB

Thus, although both Michael and Alan will connect to the same server computer with the same server user name and password (i.e., ADMINUSER/HIGHPRIV), each has his unique authentication key, which is not shareable between users or computers.

The parser program of the Secure Access System 110 will process the authentication key, decrypt it into IP address, client user name, server user ID, and server password. If the decrypted IP address and client user name match the actual client workstation and its current user, the parser program of the Secure Access system 110 uses the decrypted server user ID and server password to connect to the server computer. Otherwise, the Secure Access System 110 returns an error. If an authentication key cannot be found, the default client workstation logon user ID and password will be used.

Flow Diagrams

FIG. 3 is a flow diagram illustrating the steps performed by the Secure Access System 110 to generate and forward an authentication key. In block 300, the Secure Access System 110 obtains information comprising a server ID, a server password, a client user name, and an IP address of the client computer. In block 302, the Secure Access System 110 generates an encrypted authentication key using a defined authentication key structure and the obtained information. In block 304, the Secure Access System 110 forwards the authentication key to the user using an E-mail. The E-mail address is stored in a notification list.

FIG. 4 is a flow diagram illustrating the process of using an authentication key. In block 400, a user logs onto a client computer by entering a client user name and password. In block 402, the client computer attempts to connect to a server computer by passing the authentication key and an actual client user name and IP address of the client computer to the server computer. In block 404, the authentication key is intercepted. In block 406, the authentication key is decrypted. In block 408, the client user name and IP address obtained from the decrypted authentication key are compared to the actual client user name and IP address. In block 410, if there is a match, processing continues to block 412, otherwise, processing continues to block 414. In block 412, log on to the server computer is accomplished using the server user ID and server password obtained from the decrypted authentication key. In block 414, an error message is returned.

CONCLUSION

This concludes the description of the preferred embodiment of the invention. The following describes some alternative embodiments for accomplishing the present invention. For example, any type of computer, such as a mainframe, minicomputer, or personal computer, or computer configuration, such as a timesharing mainframe, local area network, or standalone personal computer, could be used with the present invention.

The foregoing description of the preferred embodiment of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto.